

Speckle tracking approaches in speckle sensing

Thomas O. H. Charrett^a, Krzysztof Kotowski^{a,b}, and Ralph P. Tatam^a

^aEngineering Photonics, Cranfield University, MK43 0AL, UK

^bNow at: Silesian University of Technology, Poland.

ABSTRACT

This paper reports some initial investigations into the application of feature tracking algorithms as an alternative data processing method for speckle correlation sensors capable of determining both the speckle pattern translation and rotation. The accuracy of translation measurements using the feature tracking approach was found to be similar to that of correlation based processing with accuracies of < 0.04 pixels. Rotation measurement accuracies of $< 0.05^\circ$ are found to be achievable over angle range $\pm 20^\circ$, limited by the failure to match speckles at larger rotation angles.

Keywords: Laser Speckle, Feature matching, Feature tracking, Translation measurement, Rotation measurement

1. INTRODUCTION

Laser speckle pattern correlation is a measurement philosophy that uses the translation and decorrelation of recorded laser speckle patterns to infer information about the illuminated object or the illumination source and detector. This approach can be used to measure object translation, rotation and strain¹ and recently there has been increased interest in the technique for practical applications in industry and robotics.²⁻⁴

For these applications, there is a requirement for either high speed measurement of the speckle pattern translation to enable continuous tracking of the speckle pattern, or the requirement to operate on low performance processors, for example space exploration rovers. Both of these requirements place constraints on the signal processing approach that can be used in terms of practical image size. In addition there is interest in the simultaneous measurement of the sensor in-plane rotation to provide additional information and correct for apparent translation caused by rotations.

This paper reports some initial investigations into the application of an alternative data processing method for speckle correlation sensors that mirrors the human approach used in viewing speckle patterns, i.e. identifying characteristic speckles present in both images to determine the speckle shift. Although such feature detection and tracking is a well-developed concept in computer vision their application in laser speckle sensing is limited. To our knowledge, the only prior application of feature matching is the use of the scale invariant feature transform (SIFT) algorithm for personal identification card recognition^{5,6} where a feature matching approach is necessary to ensure that speckle patterns can be matched to a database even in the presence of translations and rotations. Such approaches are also potentially well suited to speckle correlation sensing for industrial and robotics applications allowing not only the speckle translation to be determined, but also the rotation of the speckle pattern to be computed.

Although it is well known that the rotation of the speckle pattern is related to the rotation of the illuminated object however practical implementations of such sensor using this property for the detection of in-plane rotation are few. This paper investigates the potential of using feature tracking approaches used in computer vision for speckle sensing and the suitability of the different feature matching algorithms are investigated using implementations available in the open source computer vision library OpenCV.⁷ Initially a range of common feature detection, description, and matching methods are identified and their computational performance assessed in sections 2 and 3. This is followed in section 4, by a description of the geometric transformation between the

Further author information:

T.O.H Charrett: E-mail: t.charrett@cranfield.ac.uk

two sets of matched points that is used to find the translation and rotation of the speckle pattern. Finally the performance of the speckle tracking approaches for translation and rotation measurements are investigated in section 5 using simulated and experimental results.

2. FEATURE DETECTION

The first stage of the processing involved the identification of the speckles or features to be tracked, in this work some common feature detectors, with implementations available in the OpenCV⁷ library were selected for investigation and these can be grouped into three related families. The first intensity group are gradient approaches such as Harris corners⁸ and Shi-Tomasi method also known as "Good Features To Track"(GFTT).⁹

The second class of detectors are the accelerated segment tests (AST) methods including the FAST (Features from Accelerated Segment Test) detector¹⁰ and the AGAST (Adaptive and Generic Accelerated Segment Test) corner detector¹¹ which operate using a simple concept based upon the examination of pixels on a ring around the point of interest. These detectors were implemented using the OpenCV FastFeatureDetector and AgastFeatureDetector classes.⁷ The widely used combined feature detection and description algorithms, ORB¹² and BRISK¹³ use orientated versions of these FAST like detectors and these detectors were also tested and were implemented using the OpenCV ORB and BRISK classes.⁷

The final class of detectors investigated uses an alternative 'blob' detection where the image is segmented based upon the image intensity. The commonly used combined feature detection and matching algorithms Scale-Invariant Feature Transform (SIFT)¹⁴ and Speeded-Up Robust Features (SURF)¹⁵ both use a Difference of Gaussian (DoG) approximation to detect blobs, however SURF uses a more efficient algorithm and hence was chosen for this investigation. The SURF method also computes the orientation of feature although this stage can be skipped in the Upright SURF (USURF) method. Both SURF and USURF were tested using the OpenCV implementations.

2.1 Feature detection speed

Initially the speed of the detectors was investigated as this is the critical parameter to application in manufacturing and robotics where high speed processing is essential. A set of uncorrelated speckle patterns 512x512 pixel in size were acquired using a 658nm fibre coupled diode laser (FibreTec II FTEC2658) together with a camera (Ximea MQ013CG-ON) positioned at a distance of 150 mm from the surface, a cast aluminium plate. The laser output was expanded to a spot of ~8 mm diameter with the resulting speckle patterns having a speckle size diameter of ~4 pixels. This data set was then processed using the different feature detection methods described above to find between 10 and 500 features with the average frame processing times shown in Table 1.

Table 1. Average processing time per frame on an Intel core i5-4590 CPU for various methods of feature detection. Here 100 512x512 image frames were processed and typical processing times are shown for the range 10 - 500 features.

Method	Typical processing time per frame (ms)
Harris corners	8.4 - 9.0
GFTT	9.6 - 9.7
FAST	0.2 - 0.3
AGAST	0.8 - 1.0
DoG/USURF detector	30.4 - 30.9
Orientated FAST/ORB detector ^a	0.2 - 0.6
Orientated FAST/BRISK detector ^a	1.1 - 4.4
Orientated DoG/SURF detector ^a	30.6 - 40.2

^a Feature detector with orientation calculation.

The Harris corners and GFTT method show approximately constant processing times regardless of the number of features and are significantly slower than the AST detectors (FAST, AGAST, Orientated FAST/ORB, and

Orientated FAST /BRISK) but faster than methods based upon difference of Gaussian blob detection. Theoretically of the AST detectors AGAST should be fastest however the OpenCV implementation of FAST appears to be better optimised or better suited to speckle images. The additional orientation calculations in methods such as the orientated FAST methods used in ORB and BRISK, and the DoG/SURF detectors, can be seen to only increase the processing time slightly, with the ORB detector approach performing the best. From this it can be seen that the most promising feature detectors for application to speckle image processing where processing speed is significant, are the Accelerated Segment Test type detectors, such as the FAST and the orientated version found in the ORB algorithm.

3. FEATURE MATCHING

In the feature matching stage the detected features or speckle must be identified and matched between images by some form of feature descriptor. In this work four methods that have available implements in the OpenCV library were investigated for use with speckle patterns: SURF,¹⁵ USURF,¹⁵ ORB¹⁶ and BRISK.¹³

The SURF and USURF methods¹⁵ use a descriptor calculated using a square grid, rotated accordingly to the computed orientation for the SURF method. This square is divided into a 4x4 grid of sub-regions for which a 4D descriptor vector is computed based upon Haar wavelet responses, giving a feature descriptor length of 64 values. Both feature descriptors are implemented using the OpenCV SURF class.

The ORB (Orientated FAST and Rotated BRIEF) algorithm uses a BRIEF¹⁶ (Binary robust independent elementary features) like descriptor that constructs a bit string description of an image patch using a series of binary intensity tests between pixels pairs in the neighbourhood of the feature. This was implemented in the OpenCV library as the ORB feature matcher class⁷ and produces a feature descriptor of length 256 bits (from 256 intensity point-pair comparisons).

The final feature descriptor method investigated with BRISK (Binary Robust Invariant Scalable Key-points) again this uses a modified BRIEF descriptor where random pixel pairs selection has been changed to an organised procedure with points positioned on concentric circles around the key-point or feature. This was implemented using the OpenCV ORB class and produces a bit string of 512 point-pair comparisons.

After the feature descriptors have been computed then a brute force matching algorithm is employed which finds the closest matching key-point descriptor in the second set by trying each one and computing the distance based upon either the L1-norm (sum of the absolute differences $\Sigma|d_1 - d_2|$), L2-norm (sum of the squared differences $\Sigma(d_1 - d_2)^2$) where d_1 and d_2 represent the descriptor vectors to be compared. Methods that use a binary descriptor, ORB and BRISK use the Hamming distance (the number of ones in the bitwise comparison $d_1 \oplus d_2$) for fast comparison between the descriptors.

3.1 Feature descriptor and matching speed

Next the speeds of the different feature descriptor methods was investigated and the results are shown in 2. Here 500 features were identified in each of 100 frames using the FAST detector method and the average time require by each method to compute the descriptors was calculated.

Table 2. Average processing time per frame for various methods of feature descriptor for 500 features on an Intel core i5-4500 CPU.

Method	Typical description time per frame (ms)
USURF	3.3
SURF	18.7
ORB/Rotated BRIEF	1.4
BRISK	4.0

The different matching methods were also investigated and the typical times taken to match 500 features using the different error measures are shown in Table 3. It can be seen in Tables 2 and Table 3 that the binary

descriptors used in the ORB and BRISK detectors are faster, as expected and are likely most suited to high speed speckle image processing requirements.

Table 3. Average processing time per frame for the feature matching approaches for 500 features on an Intel core i5-4500 CPU.

Method	Typical description time per frame (ms)
L1-norm (64 value SURF/USURF descriptor)	9.4
L2-norm (64 value SURF/USURF descriptor)	9.4
Hamming (256 bit ORB descriptor)	5.4
Hamming (512 bit BRISK descriptor)	7.1

4. CALCULATION OF TRANSLATION AND ROTATION

The final stage in the processing is to use the features/speckles that have been matched between two images to determine the translation and rotation of the speckle pattern. That is to find the rigid transform, T , that relates a set of points P detected in the first frame to a set P' in the second, given by:

$$TP = P' \tag{1}$$

$$\begin{bmatrix} \cos \theta, & -\sin \theta, & A_x \\ \sin \theta, & \cos \theta, & A_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix} \tag{2}$$

Where A_x, A_y is the speckle translation and θ is the rotation. This can be solved provided there at least two matched points using the least squares approximation of the system:

$$\begin{bmatrix} x_1, & -y_1, & 1, & 0 \\ y_1, & x_1, & 0, & 1 \\ x_2, & -y_2, & 1, & 0 \\ y_2, & x_2, & 0, & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n, & -y_n, & 1, & 0 \\ y_n, & x_n, & 0, & 1 \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \\ A_x \\ A_y \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ \vdots \\ x'_n \\ y'_n \end{bmatrix} \tag{3}$$

5. ASSESSMENT OF THE PERFORMANCE OF FEATURE TRACKING APPROACHES

5.1 Speckle pattern translation measurement

To assess the accuracy of the feature tracking approach, artificially shifted patterns were generated using the set of speckle patterns described in Section 2. Each image was artificially translated by sub-pixel shifts, and the translation calculated using the ORB, BRISK, USURF and SURF features detection/description methods, followed by the rigid transform estimation described in equation 3. The results are shown in Figure 1 which also shows the translation calculated using the normalised cross-correlation and Gaussian peak fitting for comparison. It can be seen that the BRISK, SURF USURF feature tracking algorithms have similar performance to the normalised cross-correlation and Gaussian peak fitting method and show similar pixel-locking effects¹⁷ of $\sim 1/20$ of a pixel. The ORB algorithm performs significantly worse than the other methods with large bias errors of up to 0.15 pixels possibly due to the shorter descriptor length of 256 bits. SURF and USURF perform similarly as they are essentially the same when there is zero rotation present as the descriptor rotation for the SURF method should be approximately 0° in this test. They also show less (0.025 vs 0.033 pixels) peak-locking than the Gaussian peak fitting and normalised cross-correlation method.

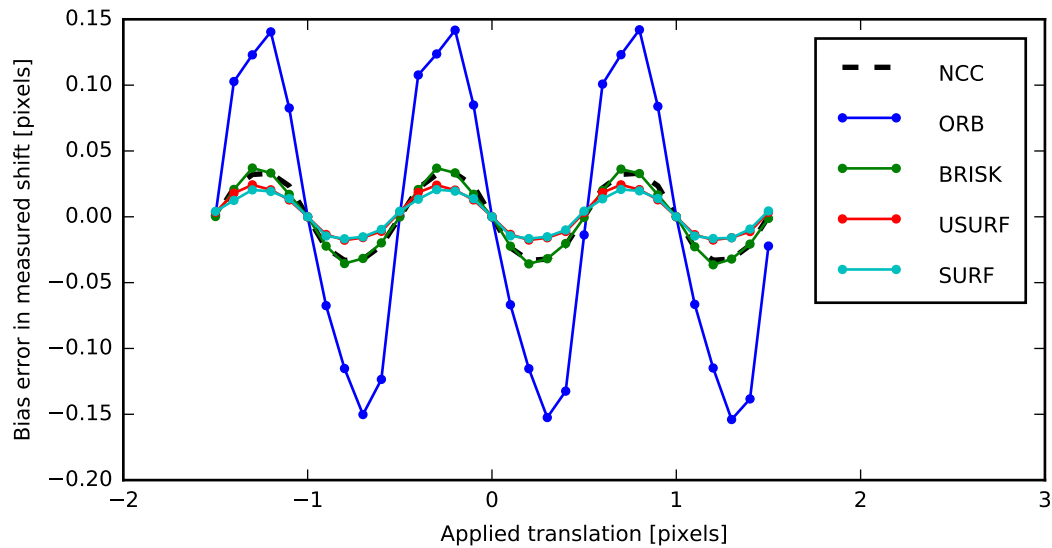


Figure 1. Accuracy of translation measurements made using the ORB, BRISK, USURF and SURF feature tracking methods, using artificially translated speckle patterns to show the remaining pixel locking effect. The normalised cross-correlation results are also shown for comparison.

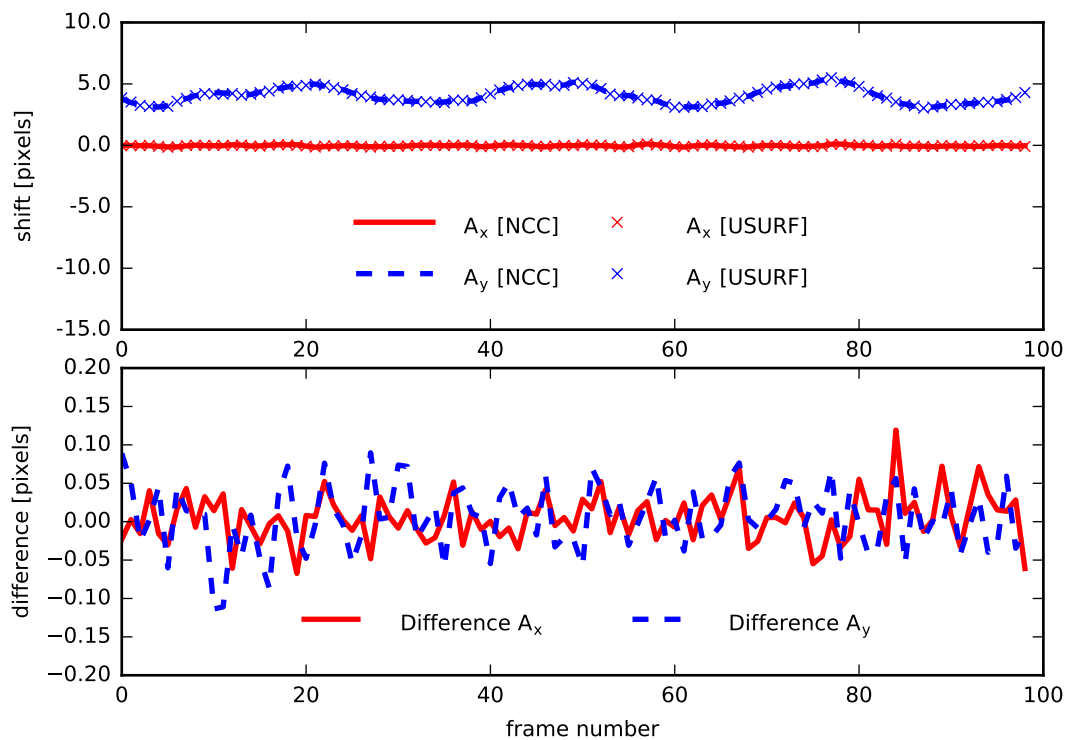


Figure 2. Comparison between experimental speckle velocimetry data processed using the normalised cross-correlation (NCC) and USURF feature tracking methods. The top axis shows the measured speckle shift while the difference between the NCC and USURF measurements are shown on the lower axis.

The performance of the feature tracking approaches was also assessed using experimental speckle patterns recorded of a speckle velocimetry sensor mounted on a translation stage moving over a surface at $\sim 5\text{mm/s}$, with

good correspondence between the correlation and feature tracking approaches found. For example, figure 2 (top) shows the results of processing using the normalised cross-correlation method and the USURF feature tracking method together with the difference between the two measurements. Here it can be seen that there is good correspondence with differences between the two approaches of < 0.1 pixels with little bias error, < 0.005 pixels, between the two and a standard deviation of ~ 0.045 pixels, suggesting that the performance of the feature tracking approach is indeed comparable to that of the conventional correlation based processing.

5.2 Speckle pattern rotation measurement

Next, the accuracy achievable for measurements of speckle pattern rotation was assessed in a similar way using a sequence of speckle patterns recorded using the experimental set-up, described in Section 2 with the surface rotated at angles between 0 and 360 degrees in 0.5 degree steps via a rotation stage (ALIO Hybrid-Hexapod AI-HYBRID-HEX-60XY-15Z-56R) with rotational repeatability ± 0.5 arc-second and 0.04 arc-second resolution. The captured images were then processed using the ORB, BRISK, USURF and SURF features detection/description methods, and the rigid transform estimation described in equation 3 used to calculate the rotation. The results are shown in Figure 3 with the top plot showing the measured versus applied rotation for the angles between 0 and 50° and the middle plot shows the error in measured angle (i.e. the difference between the applied and measured angles of rotation). From this it can be seen that there is significant differences in the performance of the different matching techniques which can be explained by the number of successfully matched features, if too few features then the error increases before there are insufficient points to solve the rigid transform. This is shown in the final plot in 3 which shows the number of feature matches versus rotation angle for each of the methods. The SURF descriptors can be seen to perform the worst, with the non-orientated version (USURF) failing first after $\pm 6^\circ$, followed by the orientated version (SURF) after $\pm 10^\circ$. The BRISK descriptor performs a little better with measurements possible up to $\pm 20^\circ$. The ORB descriptor performs the best with reliable calculation of rotation over the range $\pm 50^\circ$ and angle errors of $< 0.05^\circ$ over a range of $\pm 20^\circ$ and $< 0.02^\circ$ over $\pm 5^\circ$. This can be explained by the number of matched features where the ORB method has significantly more matches than the other methods.

6. CONCLUSIONS

In this paper a number of common feature tracking approaches have been tested for their suitability in processing of laser speckle images. These approaches appear to have potential performance similar to the cross-correlation processing conventionally used for the measurement of translation and offer the potential for the simultaneous measurement of in-plane rotation. In simulations, the BRISK, USURF and SURF feature detection methods all show similar levels of translational accuracy to the normalised cross-correlation (NCC) and 3-point Gaussian peak fit with a maximum error/peak-locking of ~ 0.04 pixels. This was also confirmed by a comparison of experimental data processed using both normalised cross correlation and feature tracking methods, with the difference between the two methods found to be < 0.1 pixel.

The main advantage of the feature tracking approach in comparison to cross-correlation based processing is in the simultaneous measurement of image rotation. Experimentally rotated speckle patterns indicate that accuracies of $< 0.05^\circ$ are achievable over angle range $\pm 20^\circ$. However the performance is very dependant upon the number of successfully matched features with the number of matched features dropping rapidly with rotation. The ORB feature description method was found to perform best in this respect, however future work should look at developing a feature descriptor that is more robust to rotation angle to allow the full range of rotation angles to be measured whilst minimising peak locking effects in translation measurements.

ACKNOWLEDGMENTS

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) UK [grant numbers EP/M020401/1, EP/N002520/1]. All data supporting this study are openly available from the Cranfield University repository at <http://dx.doi.org/10.17862/cranfield.rd.4785721>.

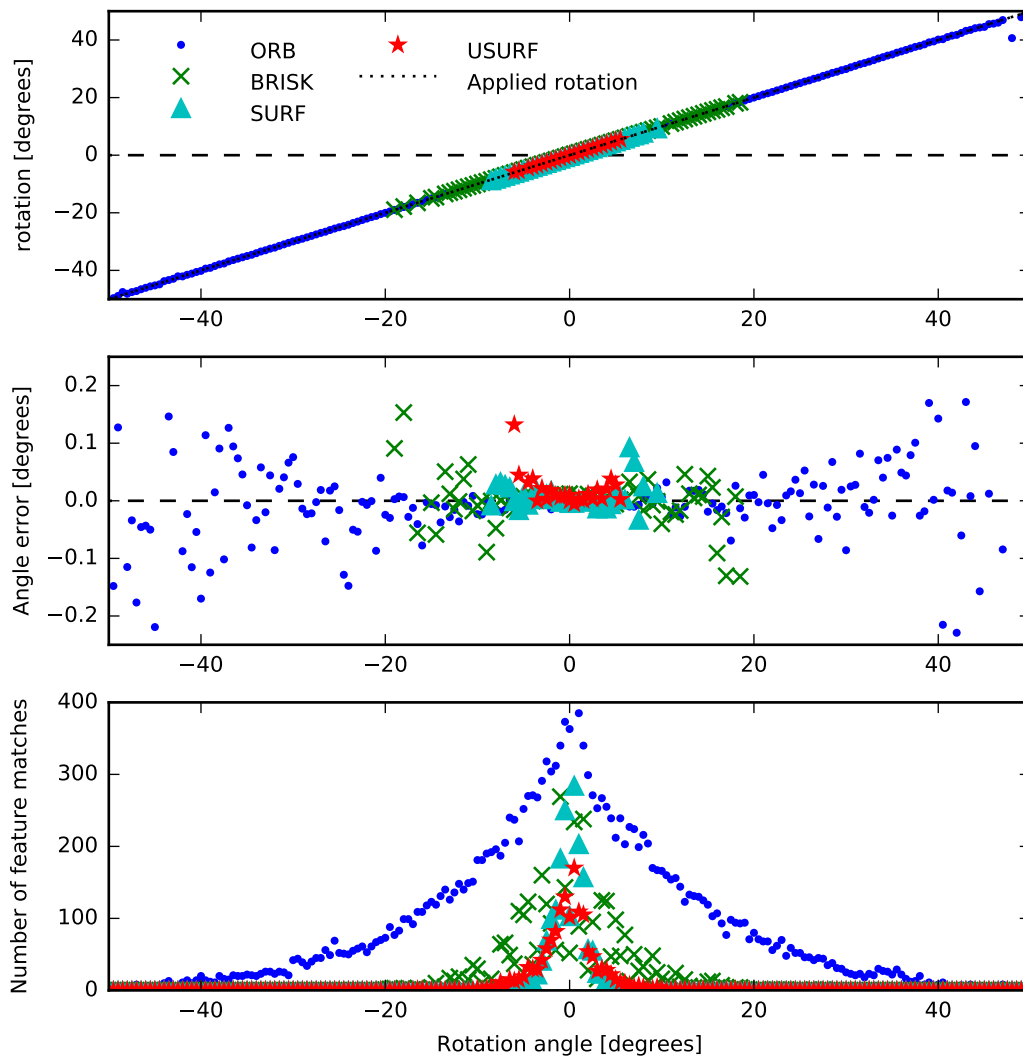


Figure 3. Accuracy of rotation measurements made using the ORB, BRISK, USURF and SURF feature tracking methods, using experimentally rotated speckle patterns. (top) measured rotation versus applied rotation, (middle) angle error, measured rotation - applied rotation and (bottom) the mean number of matched features. Missing data points are where too few features were matched.

REFERENCES

- [1] Yamaguchi, I., “Measurement and Testing by Digital Speckle Correlation,” *Proc. SPIE* **7129**, 71290Z–71290Z–9 (2008).
- [2] Farsad, M., Evans, C., and Farahi, F., “Robust sub-micrometer displacement measurement using dual wavelength speckle correlation,” *Optics Express* **23**(11), 14960 (2015).
- [3] Francis, D., Charrett, T. O., Waugh, L., and Tatam, R. P., “Objective speckle velocimetry for autonomous vehicle odometry,” *Applied optics* **51**(16), 3478–90 (2012).
- [4] Bandari, Y. K., Charrett, T. O., Michel, F., Ding, J., Williams, S. W., and Tatam, R. P., “Compensation strategies for robotic motion errors for additive manufacturing (AM),” in [*International Solid Freeform Fabrication Symposium*], (2016).
- [5] Liao, C.-M., Huang, P. S., Chiu, C.-C., and Hwang, Y.-Y., “Personal identification by extracting SIFT features from laser speckle patterns,” in [*2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*], 1341–1344, IEEE (2012).
- [6] Yeh, C.-H., Sung, P.-Y., Kuo, C.-H., and Yeh, R.-N., “Robust laser speckle recognition system for authenticity identification,” *Optics express* **20**(22), 24382–93 (2012).
- [7] “Open Source Computer Vision Library.” <http://opencv.org/>. Accessed: 2016-10-17.
- [8] Harris, C. and Stephens, M., “A Combined Corner and Edge Detector,” in [*Proceedings of the Alvey Vision Conference 1988*], 147–151 (1988).
- [9] Jianbo Shi and Tomasi, “Good features to track,” in [*Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*], 593–600, IEEE Comput. Soc. Press (1994).
- [10] Rosten, E. and Drummond, T., “Machine Learning for High Speed Corner Detection,” *Computer Vision – ECCV 2006* **1**, 430–443 (2006).
- [11] Mair, E., Hager, G. D., Burschka, D., Suppa, M., and Hirzinger, G., “Adaptive and Generic Corner Detection Based on the Accelerated Segment Test,” in [*Proceedings of the European Conference on Computer Vision (ECCV’10)*], 183–196 (2010).
- [12] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., “ORB: An efficient alternative to SIFT or SURF,” *Proceedings of the IEEE International Conference on Computer Vision* , 2564–2571 (2011).
- [13] Leutenegger, S., Chli, M., and Siegwart, R. Y., “BRISK: Binary Robust invariant scalable keypoints,” in [*2011 International Conference on Computer Vision*], 2548–2555, IEEE (2011).
- [14] Lowe, D. G., “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision* **60**(2), 91–110 (2004).
- [15] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L., “Speeded-Up Robust Features (SURF),” *Computer Vision and Image Understanding* **110**(3), 346–359 (2008).
- [16] Calonder, M., Lepetit, V., Strecha, C., and Fua, P., “BRIEF: Binary robust independent elementary features,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **6314 LNCS**(PART 4), 778–792 (2010).
- [17] Raffel, M., Willert, C. E., Wereley, S., and Kompenhans, J., [*Particle Image Velocimetry: a practical guide*], Experimental Fluid Mechanics, Springer Berlin Heidelberg, Berlin, Heidelberg (2007).

2017-05-16

Speckle tracking approaches in speckle sensing

Charrett, Thomas O. H.

SPIE

Thomas O. H. Charrett, Krzysztof Kotowski, Ralph P. Tatam, Speckle tracking approaches in speckle sensing, Optical Sensors 2017, 24 April 2017, Prague, Czech Republic, paper number 102310L
<http://dx.doi.org/10.1117/12.2264225>

Downloaded from Cranfield Library Services E-Repository